Biophysical Journal, Volume *113*

Supplemental Information

Compartmental and Spatial Rule-Based Modeling with *Virtual Cell*

Michael L. Blinov, James C. Schaff, Dan Vasilescu, Ion I. Moraru, Judy E. Bloom, and Leslie M. Loew

S1. Internal representation of compartments in VCell and use of BioNetGen and NFSim engines

We use BioNetGen and NFSim engines with augmented BNGL files to generate reaction network (BioNetGen) or perform simulations (NFSim) in multiple compartments. Fig S1 illustrates major steps of compartmental network generation:

Step 1: An augmented BNGL file is created.

Step 2: A single step of network generation (application of rules to the current set of seed species) is performed using BioNetGen engine.

Step 3: Generated set of product species is corrected to properly locate them in designated compartments.

Step 4: A corrected set of seed species is substituted into the augmented BNGL file.

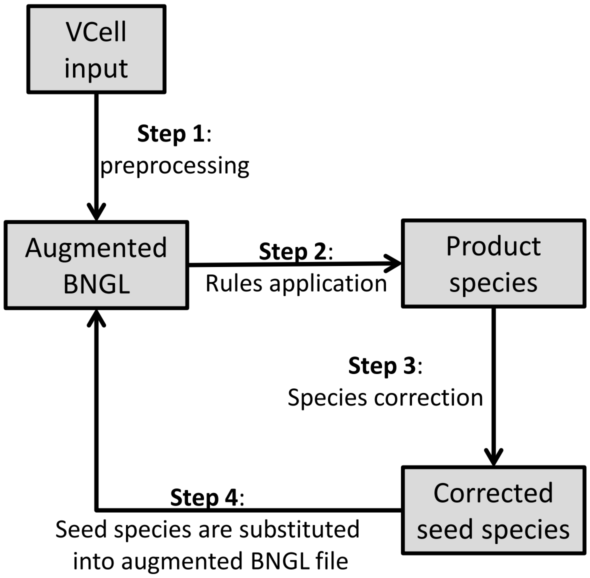

VCell input
Step 1: preprocessing
Augmented BNGL
Step 2: Rules application
Product species
Step 3: Species correction
Corrected seed species
Step 4: Seed species are substituted into augmented BNGL file

Fig S1. Block diagram for compartmental network generation.

Steps 2 -4 are repeated until no new species are generated or predefined termination condition is satisfied. Observables are not generated during these steps to speed up the process.

The last step of network generation is performed to generate a full set of reactions and observables. These sets are fixed as in step 3.

In Figure S2 we describe these steps in more detail.

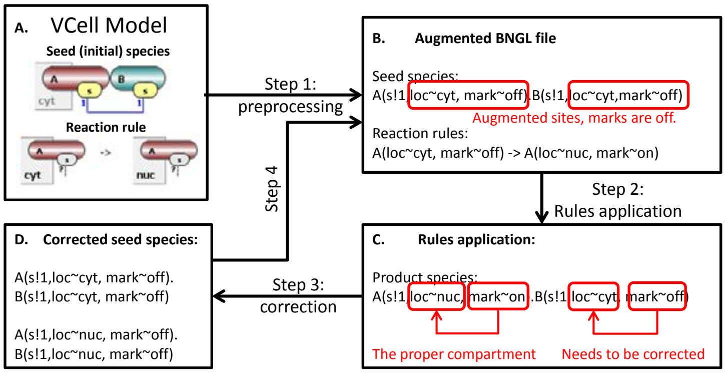


Fig S2. Example of BNGL code processing by VCell. In step 1, sites are added for compartment and mark, seed species are initialized with marks off, and reaction rule changes location and set marks on. In step 2, product species are analyzed to identify molecules with proper location (set by mark "on") and consistency with anchors. In Step 3, product species are corrected and added to the set of seed species.

First we describe the augmented BNGL file generated from VCell rule-based model (Box A). We assign two additional functional components as new sites to every BNGL molecular definition (step 1). These sites are used for internal processing only and are not visible to a user. The first added component is called "location". Its states correspond to possible locations of a molecule. Using anchoring, a modeler can optionally limit the allowable compartments for each molecule. The second added component is called “mark”. It defines whether this molecule is explicitly mentioned in a rule and is used to mark molecules that are located in the intended compartment.

User-defined reaction rules are converted into BNGL language for running with BioNetGen and/or NFSim engines. This way, the file can be processed by both BioNetGen and NFSim (NFSim currently does not support cBNGL). In the augmented BNGL file, changes in “location” and “mark” sites from reactant to product species patterns is defined in every reaction rule (Box B).

VCell defines each reactant and product species patterns to be located in a given compartment. Thus, a “location” site is changed from reactant pattern’s compartment to product pattern’s compartment for all molecules explicitly mentioned in a rule definition (Box B).

A “mark” site is changed for all molecules explicitly mentioned in a rule definition from “off” to “on” (Box B). This site is required because a species generated by the rule may contain multiple molecules, including those not explicitly mentioned in the rule definition but are bound to a molecule explicitly present in a reactant or product patterns. These molecules may have contradictory location sites, because a rule does not have any information to change location for molecules not explicitly defined in the rule definition. Thus, the component “mark” is used to resolve this conflict by identifying those molecules (with mark site in “on” state) that are located in the intended compartment. This compartment is then assigned to all molecules in a product species (Box C).

Below we summarize steps of the compartmental network generation using BioNetGen in greater detail. All processing but step 3 is done using Java code in VCell.

1. An augmented BNGL file is created:
    a. All seed species are initialized with “mark” site in “off” state.
    b. Every reversible reaction rule is broken into two irreversible rules.
    c. Every irreversible reaction rule is augmented to change location site and the state of the “mark” site from “off” to “on” in molecules explicitly mentioned in product pattern.
2. A single step of network generation (application of rules to the current set of seed species) is performed using BioNetGen engine.
3. Generate a new set of seed species:
    a. Product species are corrected by changing “location” site of all molecules to the intended location site specified in the rule (identified as a location site of molecules that have a “mark” site in “on” state).
    b. Product species that have molecules in anchored compartments override location in a rule, so such species are located in anchored compartment. VCell user interface assures that there will be no conflict in anchors during model specification.
    c. Duplicated product species are removed.
    d. A new set of seed species is compiled consisting of the previous set of seed species and product species not isomorphic to existing species.
    e. All “mark” sites in the seed species are set to “off”.
4. A new set of seed species is substituted into the augmented BNGL file.

5. Steps 2 -4 are repeated until no new species are generated or predefined termination condition is satisfied. Observables are not generated during these steps to speed up the process.
6. The last step of network generation is performed to generate a full set of reactions and observables. These sets are fixed as in step 3.

To use NFSim for network-free simulation, the same augmented BNGL file is generated (Step 1 above) and converted to XML input for NFSim. NFSim code is updated to correct product molecular constructs and remove duplicates during each simulation time step. Post-processing of species generated by iterations of BioNetGen (Step 2 above) is replaced by post-processing of molecular complexes after each rule firing by NFSim. The NFSim *MoleculeType* class was extended with *bool bHasAnchors* and a vector of anchor names. VCell uses NFSim with bookkeeping enabled, thus complexes are processed at each time point using the same steps as for network generation:

a. After each reaction event, each product complex is searched to identify molecules with “mark” site state set to "on".
b. Find location site state for marked molecules (this is the desired location) and any anchors present in that complex.
c. Reconcile locations of “marked” molecules with allowed locations defined by anchors.
d. For all molecules in the complex, set location site state to the desired location and set all marked site states to "off."

When exporting as BNGL, we use standard cBNGL conventions, as described in supplemental material S2.

S2. Rule-based model specification in VCell vs cBNGL

VCell can import and export cBNGL. However, mapping is not one-to-one.

Below is cBNGL export of the Ran_tutorial_RB model used in the paper.

```
begin model

begin compartments
EC        3        1
cyt       3        1
nuc       3        1
pm        2        1
nm        2        1
end compartments

begin parameters
end parameters

begin molecule types
Ran(cargo)
C(site,Y1~u~p,Y2~u~p,Y3~u~p)
RCC1(site)
```

```
end molecule types

begin anchors
RCC1(nuc)
end anchors

begin seed species
@nuc:Ran(cargo!1).C(site!1,Y1~u,Y2~u,Y3~u) 4.5E-4
@nuc:RCC1(site) 4.5E-4
end seed species

begin observables
Molecules Ran_cyt @cyt:Ran()
Molecules Cargo_cyt @cyt:C()
Molecules RCC1_nuc @nuc:RCC1()
Molecules Cargo_phosp_cyt_total @nuc:C(Y1~p!?) @nuc:C(Y2~p!?) @nuc:C(Y3~p!?)
Molecules Cargo_nuc @nuc:C()
Molecules Cargo_phosp_cyt @cyt:C(Y1~p!?,Y2~p!?,Y3~p!?)
Molecules Ran_bound_cyt @cyt:Ran(cargo!+)
end observables

begin reaction rules
Transport:  @nuc:Ran(cargo!+) <-> @cyt:Ran(cargo!+)                    2.0 * 602.0, 0.0
Ran_C_bind_cyt:     @cyt:Ran(cargo!1).C(site!1) <-> @cyt:Ran(cargo) + @cyt:C(site)              1.0, 100.0
C_p1:       @cyt:C(Y3~u!?) <-> @cyt:C(Y3~p!?)                              10.0, 1.0
C_p2:       @cyt:C(Y2~u!?) <-> @cyt:C(Y2~p!?)                              10.0, 1.0
C_p3:       @cyt:C(Y1~u!?) <-> @cyt:C(Y1~p!?)                              10.0, 1.0
Ran_RCC1_bind:      @nuc:Ran(cargo) + @nuc:RCC1(site) <-> @nuc:Ran(cargo!1).RCC1(site!1)                1.0, 100.0
Ran_C_bind_nuc:     @nuc:Ran(cargo!1).C(site!1) <-> @nuc:Ran(cargo) + @nuc:C(site)              1.0, 100.0
end reaction rules

end model

generate_network({max_iter=>10,max_agg=>10,overwrite=>1})
```

To import it into BioNetGen, the following corrections are needed:

1. A compartments block has to be modified to specify enclosing compartments and the hierarchy EC-> pm -> cyt -> nm -> nuc:
   ```
   begin compartments
   EC      3       1
   pm      2       1     EC
   cyt     3       1     pm
   nm      2       1     cyt
   nuc     3       1     nm
   end compartments
   ```
2. Anchors block to be removed.
3. Check units consistency. BioNetGen does not import units, and only requires that they are consistent. VCell operates with units that may be distinct on membrane and in volumetric compartments. In particular, in the example above rates for mass-action kinetics for all volumetric reactions are expressed in per second (first order) and per second per micromolar

(for the second order reactions), while a transport reactions on a membrane are expressed surface densities, so the rate is in molecules per um^2 per second per micromolar.

After these modifications, the file can be processed and generates 37 species and 100 reactions, as if VCell model when RCC1 is not anchored. To make a cBNGL model identical to the VCell, additional specifications are needed that are beyond the scope of this paper.

The import of cBNGL files into VCell requires few modifications (errors are displayed in a pop-up window while importing):

1. In species block, change Molecule @compartment to @compartment:molecule syntax
2. By default, universal rules (rules without explicit compartments defined) will be created in the first compartment (by specification sequence in the BNGL file). Such rules need to be manually duplicated among other required compartments (VCell provides a button for a duplication operation).
3. In reaction rules and observable blocks, change species_pattern @compartment to @compartment:species_pattern syntax, having one compartment per species pattern. Molecules in species patterns cannot be assigned to different compartments, as admissible in cBNGL.

Quick Start Guide:
Create & Simulate Rule-Based Models in VCell 6.1

VCell installation and set-up

Go to http://www.vcell.org/ , select “DOWNLOAD VCELL” from the top menu, choose the proper OS and click the link to VCell **6.1**.
Internet-dependence: After installation, you can run the local VCell application without an Internet connection and without logging in to the model database; in this case, however, you will only be able to work with local files being saved by using File > Export. Whenever you are connected to the Internet, VCell will automatically check for and download any updated version.

Navigation through VCell Modeling Framework

The Virtual Cell opens with four panels.

- The **upper left panel** of VCell provides navigation links to different steps of the VCell modeling process that become active in the **top right panel**.
- Select any element in the top right panel and you can view/edit its **Object Properties** in the **bottom right panel**.
- The **bottom left panel** shows models stored in VCell databases and other resources.

Quick tour of rule-based modeling

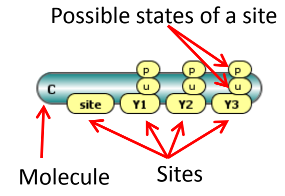
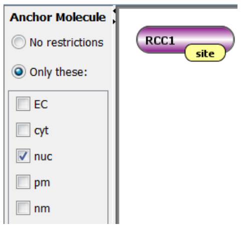


Fig. 1. Molecule

Rule-based modeling involves the representation of **species** as structured objects consisting of **molecules** and molecular interactions as **reaction rules** for transforming the attributes of these objects. It allows one to systematically incorporate site-specific details about molecular interactions into a model.

Molecules (Fig. 1) are the principal component of rule-based model specification. Molecules are comprised of sites that can bind to each other, both within a molecule and between molecules. Sites typically represent physical parts of proteins, such as extracellular and trans-membrane domains or phosphotyrosines of a receptor. Sites may also be associated with a list of states, intended to represent states or properties of the site, e.g. phosphorylation status.



Anchor Molecule
○ No restrictions
● Only these:
□ EC
□ cyt
☑ nuc
□ pm
□ nm

Fig 2. Anchoring molecules

Molecule can be anchored to specific locations (Fig. 2), so all species that contain this molecule will be located in these locations only.

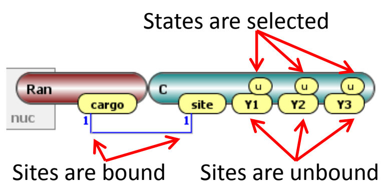


Fig 3. Species comprised of multiple molecules

Species (Fig. 3) are composed of molecules with bonds connecting binding sites. If a Molecule has multiple states, each state must be specified - otherwise the species will present a pool of different chemical entities and will not be valid.

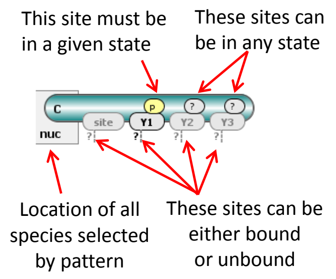
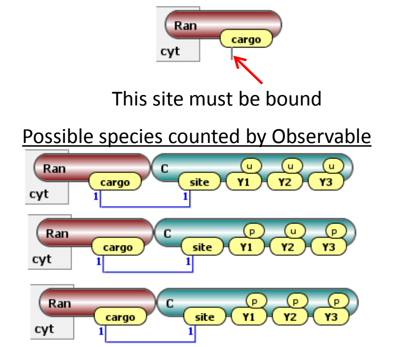

This site must be in a given state
These sites can be in any state
C
nuc
p
?
site
Y1
Y2
Y3
Location of all species selected by pattern
These sites can be either bound or unbound

Fig 4. Species pattern

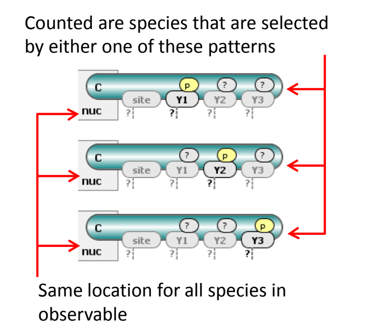
Species Patterns (Fig. 4) specify a set of possible species to be selected as participants in reaction rules and in observables. Patterns are comprised of molecules. The states of sites may be left unspecified; thus a pattern may select multiple species. Moreover, binding sites may have implicit binding status (has external bond or may be bound) where its binding partner is not explicitly defined. Such patterns may be inclusive of species that contain molecules not explicitly specified in a pattern but being possibly bound to molecules within it.



This site must be bound

Possible species counted by Observable

Fig 5. Observable and selected species

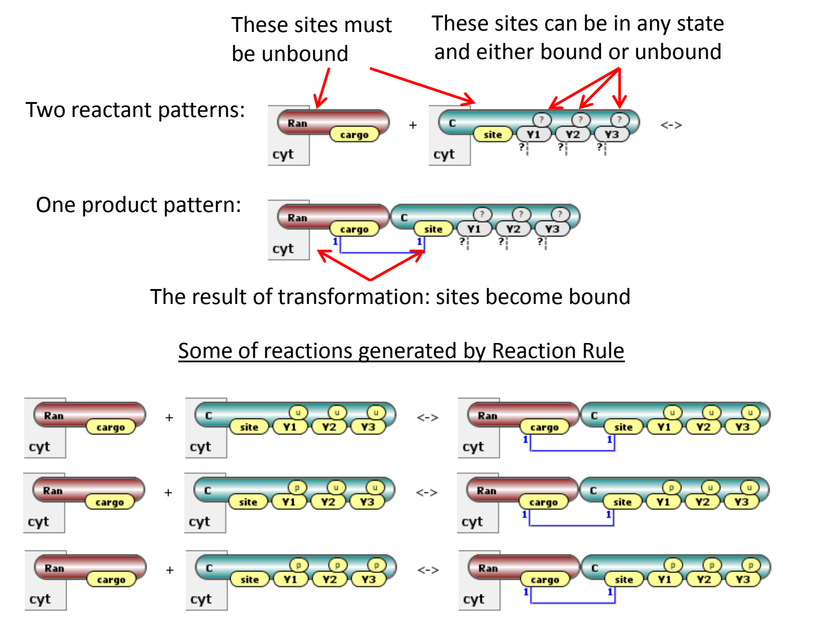


Counted are species that are selected by either one of these patterns

Same location for all species in observable

Fig 6. Observable consisting of several species patterns

Observables (Fig. 5) are used to specify model outputs, which are functions of the population levels of multiple chemical species that share a set of properties. For example, to simulate the tyrosine phosphorylation level of a particular protein, one needs to specify an Observable that is a function for the total amount of all chemical species containing the phosphorylated form of this protein. Observables are computed over a set of chemical species (Fig. 6) that share similar properties.



These sites must be unbound

These sites can be in any state and either bound or unbound

Two reactant patterns:

One product pattern:

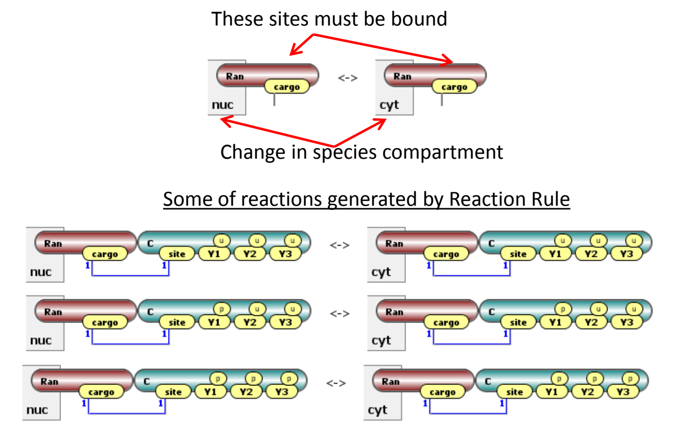
The result of transformation: sites become bound

Some of reactions generated by Reaction Rule

Fig 7. Reaction rule for binding molecules.

Reaction rules define transformation of multiple species at once (Fig 7, 8). Species to be transformed are selected by reactant pattern(s). Product pattern(s) define the end result of transformation. Product pattern may differ from reactant by adding, deleting or re-assigning of molecules, adding or deleting bonds and changing site states.

Network generation: given an initial set of species with non-zero amounts (**seed species**) and a set of reaction rules, every rule in the set is applied to those species among the seed species that can be selected as reactants, generating proper well-defined reactions and an extended set of species (seed species and species that are products of these reactions). The set of rules is again applied to this extended set of species to generate even more reactions and more species. This process is iteratively repeated until no new species and reactions are generated.



These sites must be bound

Change in species compartment

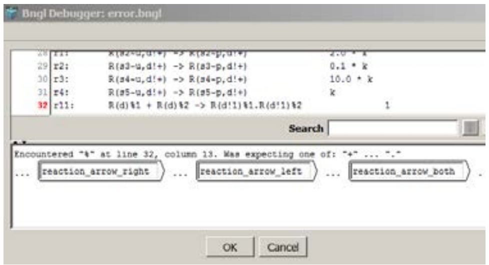
Some of reactions generated by Reaction Rule

Fig 8. Reaction rule for transport.

BioNetGen (**Bio**logical **Net**works **Gen**erator, http://pubmed.org/19399430) and **NFSim** (**N**etwork-**F**ree **Sim**ulator, http://pubmed.org/21186362) are two back-end engines that are used in VCell 6.0 to work with rule-based models.

Load an existing rule-based model

Public rule-based models are located in **Tutorial VCell 6.1 (Rule-Based)** in bottom left panel under **VCellDB -> BioModels**. These models are accessible when users log in to VCell from the Internet.



```
28 r1:     R(s2~u,d!+) -> R(s2~p,d!+)          2.0 * k
29 r2:     R(s3~u,d!+) -> R(s3~p,d!+)          0.1 * k
30 r3:     R(s4~u,d!+) -> R(s4~p,d!+)          10.0 * k
31 r4:     R(s5~u,d!+) -> R(s5~p,d!+)          k
32 r11:    R(d)%1 + R(d)%2 -> R(d!1)%1.R(d!1)%2        1
```

```
Encountered "%" at line 32, column 13. Was expecting one of: "+" ... "."
... reaction_arrow_right ... reaction_arrow_left ... reaction_arrow_both .
```

Fig 9. Correcting errors in VCell import

If you have a BNGL code, you can choose **File -> Import**. The code will be imported and a rule-based model will be created with two applications: Network-Free called NFSim, and deterministic network application called BioNetGen. If some features of BNGL file are not supported by VCell, a pop-up window is launched with a suggestion to correct unsupported features (Fig 9). When some features in BNGL file (such as fixed value of concentration) are supported by BioNetGen but not NFSim, only a single deterministic rule-based application will be created. You will be asked about units and simulation volume during import.

A model can be saved to the database (**File > Save As)** or locally **(File->Save As Local)**.

Create a new rule-based model

Please note that **Help->Help** menu provides a very detailed searchable description of all steps in modeling, definitions and specifications. This guide is intended to give just a short overview of modeling process.

1. **Create one or more Molecules** (Fig. 10).
2. **Create one or more Species** (Fig 11).

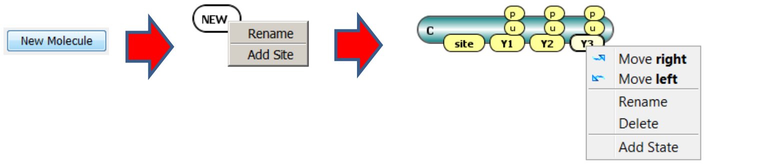


Fig. 10. Molecule specification in VCell. If necessary, anchor molecule to compartments.

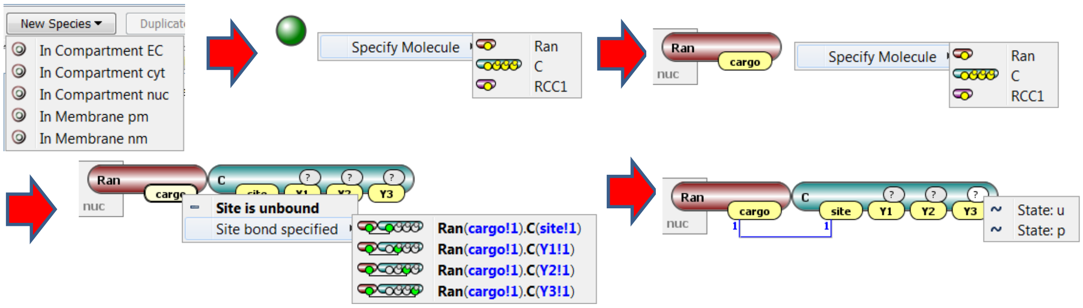


Fig. 11. Species specification in VCell.

3. **Create one or more Observables** (Fig 12).

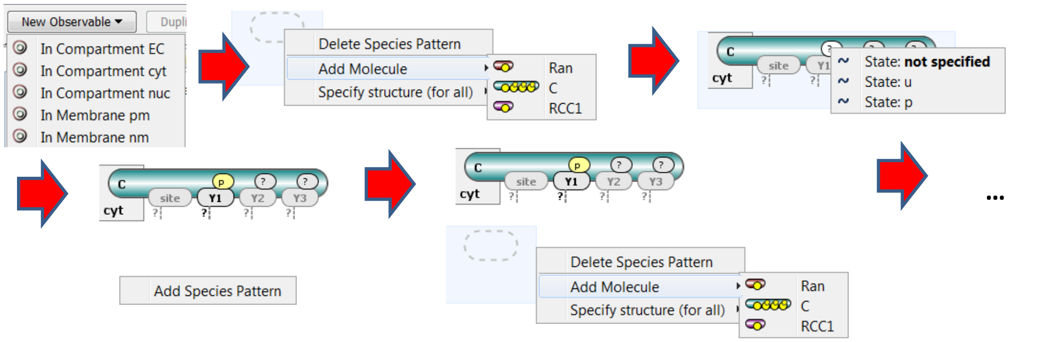


Fig 12. Specification of Observable in VCell

4. **Create one or more Reaction Rules** (using **Editor** panel in **Reactions**, Fig. 13).

After reaction rule is specified, click on **Kinetics** tab and set whether the rule is reversible, forward and reverse microscopic rate constants for default mass action kinetic law. By default, all reactions generated by a reaction rule have a mass-action kinetic law with forward and reverse rate constants specified here. However, these constants can be adjusted by statistical and symmetry factors (see “Expert Options” section below).

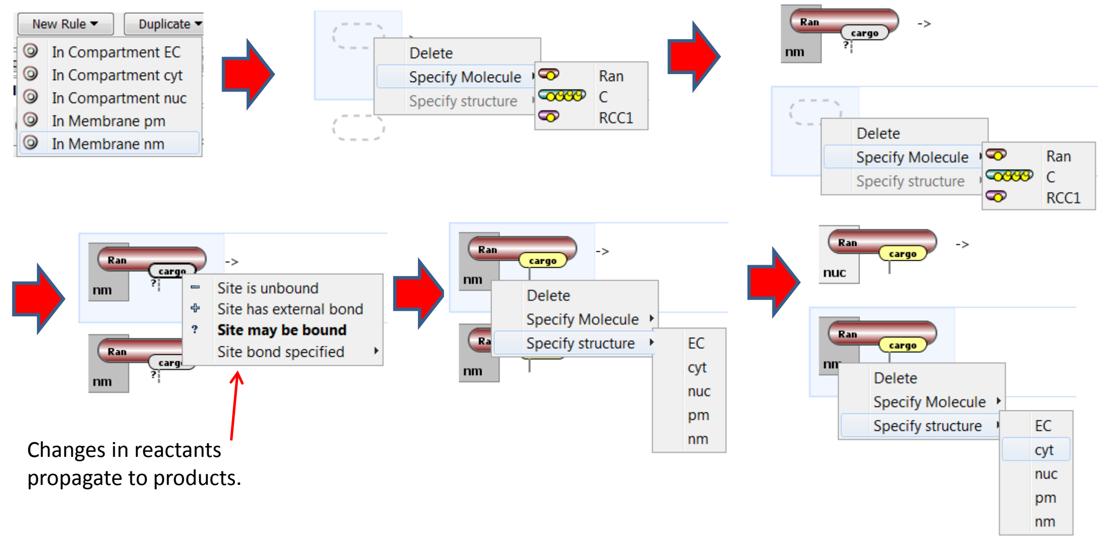


Fig 13. Specification of reaction rule for a transport reaction rule

Simulate rule-based model

Once rule-based features are specified in **Physiology**, a user has multiple choices of how to simulate. As standard with Virtual Cell, a Rule-Based BioModel may have numerous **Applications**, including **Deterministic**, **Stochastic** and **Network-Free** applications. Each Application, in turn can have multiple **Simulations**, in which different numerical values or simulation conditions are used.

- **Deterministic** and **Stochastic applications** rely on BioNetGen to create the reaction network. Network specification parameters (**Maximal Number of Iterations** to apply rules and **Maximal Number of Molecules per Species** in generated network) can be used to control the size of the network, which otherwise can become infinite or too large. After the network is generated, regular deterministic or stochastic simulators can be run.
- **Network-free Application** simulation avoids network generation and directly simulates the amounts of model observables using NFSim engine. Rather than generating and tracking all possible chemical species, network-free approach follows only the molecular configurations that exist at a given time.

Deterministic Non-spatial Applications: d/dt

- Select Application in the upper left panel, right click **-> New Application -> Deterministic**. By default, new Application is a single compartment.
- Go to **Specifications -> Species** to specify initial values of species and fixed values (clamping).
- Go to **Specifications -> Reactions** to enable/disable some of reaction rules.
- Go to **Specifications -> Network** to check how a reaction network looks like for different constraints on network generation. **Apply** constraints after you’re happy with the network.
- Go to **Specification->Simulations** to set simulation parameters and run.
    - Create and manipulate Simulations using the icons on the left to **Add New**, **Copy**, **Edit**, **Delete**.

  - Click **Edit** to check solver settings. Use the **Parameter**s tab in the edit simulation window to create Simulations with specific parameter values being overridden.
  - **Green triangle** on the right sends simulation to the server. Simulation results will be available after simulation is completed. Logging in is required. After green button is pressed, any other work with VCell may be continued, or a user can close VCell completely. Simulation results will be saved in the database.
  - **Blue triangle** on the right performs simulations on a client's computer. No other VCell operations are possible till simulation is completed.
  - Simulation results can display Observables, Species (that were specified in Physiology), Generated Species (those generated in the network – their structure can be seen in **Specification -> Network**) and Other – which include reaction fluxes and user-defined functions.
  - Ctrl and Shift can be used to select multiple outputs.
  - Simulation Results for "green" run can be viewed in a separate window invoked by clicking on the **Results** icon on the left. Results can be viewed while a Simulation is still running; the data displayed will update automatically at the same time with the Simulation status.

Deterministic Spatial Applications:

- Most settings are the same as for deterministic application, plus add extra specifications:
- Go to **Specifications -> Geometry** to specify 2D or 3D geometry and map it to compartments.
- In **Specifications -> Species** define diffusion constant for seed species. All newly generated species will have the same diffusion constants.
- Go to **Specification->Simulations** to set simulation parameters and run.
  - All parameters same as in deterministic non-spatial.
  - Under Edit -> Mesh, define Mesh size for spatial simulations, under Edit -> Solver specify solver settings.

Stochastic Non-spatial Applications:

- Most settings are the same as for deterministic non-spatial.
- **Specifications -> Species** will give a value of either concentrations, or particle numbers. Simulations will be performed in particles. Conversion is done using compartment volume that is defined in **Specifications->Geometry**. Make sure simulation volume is appropriate to avoid very large number of particles.
- Simulation results can be seen in concentrations and number of particles (name_Count).

Network-Free Applications:

- Most settings are the same as for stochastic non-spatial application.
- This is a stochastic application, so compartment volume must be defined in **Specifications->Geometry**
- **Specifications -> Network-Free** can be used to generate a rule-based model from a regular reaction network. For a purely rule-based model pressing this button will generate a new rule-based model identical to the existing. Check "Expert Options" section for more details.
- NFSim simulator has special settings – check them carefully under **Simulation -> Edit->Solver**.

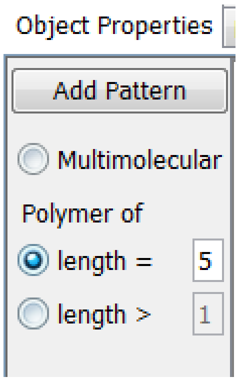
Stochastic spatial Applications:

- Same as for stochastic non-spatial.
- Go to **Specifications -> Geometry** to specify 2D or 3D geometry and map it to compartments.
- In **Specifications -> Species** define diffusion constant for seed species. All newly generated species will have the same diffusion constants.

- **Specifications -> Species** will give a value of either concentrations, or particle numbers. Simulations will be performed in particles. Conversion is done using compartment volume that is defined in **Specifications->Geometry**. Make sure simulation volume is appropriate to avoid very large number of particles.
- Simulation results can be seen in concentrations and number of particles (name_Count).

Working with polymers

- If a reaction rule has multiple identical molecules, they are enumerated to provide one-to-one mapping from reactants to products. The match is established automatically, but can be reassigned with a right click on a molecular shape.
- Observables can be specified as multimolecular complexes as in Figs 5 & 6. Observables that count species comprising of multiple identical molecules are defined as polymers (Fig. 14).



○ Multimolecular

Polymer of

● length = 5

○ length > 1

Fig 14. Defining polymeric observable that consist of exactly 5 molecules of a given type.

Expert Options

- Every VCell BioModel can be enhanced by creating Molecules and assigning molecular composition to species.
- One can mix reactions and reaction rules (see *Mix_Reactions_Rules* model in Tutorial folder). In particular, species that are used as initial seed species for rule-based models can be used as species in reaction networks.
- Only mass-action kinetics is supported in reaction rules. In deterministic applications, one can put expressions depending on Species as forward and reverse rates (see *RB_Enzyme_Kinetics* model in Tutorial folder). In Stochastic and Network-Free Applications, only numerical expressions are allowed.
- Using Network-Free Application, please be aware:
    - While it can be used for every VCell BioModel obeying the limitations above, it is designed to be used to simulate rule-based models.
    - This is a stochastic simulation, so simulation results will vary by run.
    - It operates with particles, so using “copy as” to convert a deterministic application to a Network-Free application may lead to a very large number of particles. This number can be adjusted by changing simulation volume in **Specifications -> Geometry**. The default maximal number of particles is 200,000 (it may be increased in **Simulations -> Edit -> Solver**)
    - The user is unable to see what species are populated during a simulation. To test if reaction rules produce expected species and reactions, the user is advised to generate a deterministic rule-based application and test network properties.
- Using Deterministic and Stochastic Applications, one can create a new VCell BioModel that consists of all generated species and reactions; species have molecular details, while reactions carry rule name under reaction name. Note that reaction rates in reaction network are adjusted for symmetry factors and statistical factors:
    - The rate law associated with a rule is a microscopic rate for reactions of the form A+B -> products(). If a rule generates a reaction of the form A+A -> products(s), a reaction rate for such symmetric reaction is multiplied by a **symmetry factor** of 1/2.
    - In some cases, a reaction generated by a rule can occur in multiple ways that are indistinguishable. For example, a rule A.A -> A.A’ can be applied in two different ways: either the first A wil be modified, or the second A will be modified. In these cases, the single-site rule rate law is multiplied by a statistical factor 2 to obtain the rate of the reaction.

- For any VCell BioModel that satisfies above constraints (mass-action only) one can create a rule-based model by creating Network-Free Application and going to **Specifications -> Network-Free -> Create new Rule-Based VCell BioModel.** If no rule-based elements were present, new molecule will be created for every species and trivial rules corresponding reactions will be done. If species have molecular structures, they will be used in rule-based model. Note that the same reaction expression treated as a reaction and as a reaction rule will have different rates:
    - VCell will take care of reactions of the form A + A -> product(s), and the rate of generated reaction rules will be multiplied by 2 to account for symmetry factors.
    - VCell does not take care of reactions where reactants have symmetric sites and reaction rule will have a statistical factor. If this is not properly treated by a modeler, the resultant rule-based model may show different simulation results than its network precursor.

Limitations compared to stand-alone tools (temporary, will be lifted in future releases of VCell). (for BioNetGen/NFSim experts)

- Only mass-action kinetic laws are supported. No functions are supported (only expressions in forward and reverse rates). Total rate is not supported.
- Molecules cannot have identical sites.
- No exclude/include operations are supported.
- All reaction rules with identical molecules are automatically assigned matches between reactant and product patterns.
- Matches between molecules in reactants and products in BNGL file are not imported, but can be reassigned within VCell interface.
- To speed-up simulations, the number of generated species cannot exceed 800 and the number of reactions should be less than 2,000. If network constraints provide larger numbers, they will not be accepted for deterministic and stochastic simulations.